Bab 2

Contoh VBA Praktis untuk Bisnis Sederhana

Ada banyak unsur di dalam VBA yang bisa digunakan sebagai tool untuk mengelola bisnis kita secara sederhana. Ada object, methods, property, dan sebagainya yang sudah dibahas sebelumnya. Selain itu, ada juga sub procedure dan function. Ketika bicara tentang masalah hitung-hitungan atau manipulasi data, besar kemungkinan sub procedure dan function itu dipakai terus menerus. Oleh karena itu, sebagian besar fokus perhatian kita terletak pada cara pembuatan sub procedure dan function.

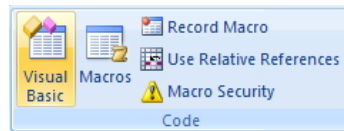
Secara umum, jika dikaitkan dengan bisnis sederhana, maka function dan sub procedure dapat didefinisikan sebagai berikut:

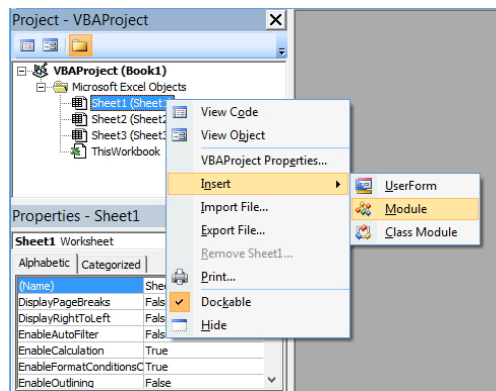
- Function berfungsi untuk "menghasilkan sesuatu dari penghitungan-penghitungan yang dihitung menggunakan rumus yang kita ciptakan sendiri."
- Sub Procedure berfungsi untuk "melakukan perintah tertentu" sesuai dengan script yang kita buat.

Kalau Anda ingin menulis formula atau fungsi di dalam Formula Bar, maka ciptakanlah function. Sedangkan kalau ingin membuat tindakan tertentu, misalnya menginput data atau membuat user form, maka umumnya diciptakan menggunakan sub procedure.

Membuat Sub Procedure Sederhana

Dengan menggunakan sub procedure, Anda bisa membuat berbagai macam perintah. Sebagai contoh, Anda bisa membuat semacam kotak dialog di dalam jendela MS Excel untuk keperluan-keperluan tertentu. Berikut salah satu contoh sub procedure yang bisa Anda buat.

1. Klik tab **Developer** pada jendela MS Excel.
2. Setelah itu, pilihlah ikon **Visual Basic** yang ada di sisi kiri ribbon Developer itu, tepatnya berada di dalam kelompok **Code**.

Memilih ikon Visual Basic

3. Lantas, klik **Sheet1(Sheet1)** pada panel **Project – VBAProject**.
4. Klik-kanan dan pilih menu **Insert > Module**.

Pilihlah opsi Module untuk membuat script VBA yang baru

5. Tulislah script seperti berikut ini di dalam jendela **Code**:

```
Sub Halo()
 Dim Sapa As String
```

```
  Sapa = "Halo, Apa Kabar!"
  MsgBox Sapa
End Sub
```

6. Jalankan sub procedure di atas dengan menekan tombol **F5** dan lanjutkan dengan menekan tombol **Run** jika perlu.
7. Nanti akan terlihat tampilan kotak dialog seperti gambar di bawah ini. Tekan tombol **OK** untuk menutup kotak dialog itu.

Kotak dialog yang menampilkan teks seperti kita tulis dalam script VBA

> ***TIGA CARA MENULIS SCRIPT VBA***
>
> *Ketika nanti Anda menulis script VBA, maka Anda bisa memilih salah satu dari ketiga metode penulisan di bawah ini:*
>
> - *Menulis tiap perintah dalam baris-baris terpisah. Ini adalah cara yang paling lazim di mana tiap-tiap perintah dibuat dalam baris-baris yang berbeda. Konsekuensinya, script akan memanjang ke bawah sehingga harus selalu melakukan scrolling ke arah bawah. Pada contoh script di atas, Anda sedang menulis script dengan perintah yang ditulis dalam baris-baris terpisah.*
> - *Menulis perintah dalam satu baris yang panjang. Untuk memisahkan satu perintah dengan perintah lainnya, gunakanlah tanda baca titik-dua (:). Metode ini praktis karena tidak memakan banyak baris. Namun cukup membingungkan jika Anda tidak terbiasa dengan cara seperti ini. Anda bisa menulis script di atas lewat satu baris sebagai berikut:*

```
Sub Halo()
  Dim Sapa As String: Sapa = "Halo, Apa Kabar!": MsgBox Sapa
End Sub
```

- *Satu baris perintah yang panjang, dipecah dalam beberapa baris. Metode ini ditempuh ketika dalam satu baris yang panjang, Anda memecahnya menjadi beberapa baris. Untuk memecahnya, gunakan tanda (_) seperti contoh berikut ini:*

```
Sub Halo()
  Dim Sapa _
  As String
  Sapa = _
  "Halo, Apa Kabar!"
  MsgBox Sapa
End Sub
```

Ketiga script di atas jika dijalankan, akan menghasilkan output yang sama.

Contoh-Contoh Sub Procedure Praktis untuk Terapan

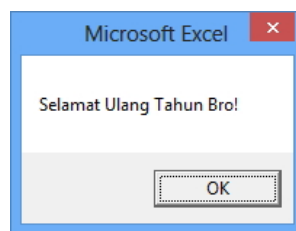
Anda bisa membuat berbagai macam sub procedure praktis untuk terapan bisnis, seperti mengetahui tanggal ulang tahun, membuat kata sandi sederhana, dan lain sebagainya. Berikut contohnya:

Sub Procedure untuk Menampilkan Kotak Dialog Jika Saat Ini Sedang Berulang Tahun

Tulislah script di bawah ini untuk menampilkan kotak dialog yang akan muncul secara otomatis jika tanggal saat ini sama dengan tanggal yang telah diatur di dalam script VBA itu:

```
Sub UlangTahun()
  If Date = #6/18/2013# Then
```

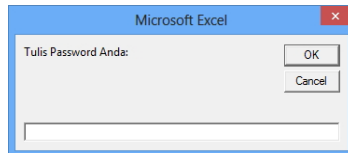
```
 MsgBox "Selamat Ulang Tahun Bro!"
 End If
End Sub
```

Contoh script di atas memanfaatkan kondisional If-Then untuk mengecek apakah hari ini tanggal 18 Juni 2013 atau tidak. Jika ya, maka akan muncul kotak dialog seperti gambar di bawah ini:

Kotak dialog berisi ucapan yang akan muncul ketika script di atas dijalankan

Sub Procedure untuk Membuat Username Sederhana

Anda bisa membuat sub procedure untuk memproteksi worksheet. Ide dasarnya, Anda bisa membuat InputBox di dalam script untuk menampung data isian. Selanjutnya, data isian itu dicocokkan dengan kata kunci yang telah kita tentukan. Berikut contoh penulisan script yang melibatkan InputBox tersebut:

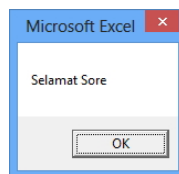
```
Sub CekPassword()
 UserName = InputBox("Tulis Password Anda: ")
 If UserName <> "Jubilee" Then GoTo Salah
 MsgBox "Login Berhasil. Selamat Datang Jubilee"
 Exit Sub
Salah:
 MsgBox "Maaf. Password yang Anda masukkan salah"
End Sub
```

Jalankan script di atas dengan menekan tombol **F5**. Selanjutnya akan muncul kotak dialog seperti gambar di bawah ini. Tulislah kata kunci yang sudah Anda atur di dalam script.

Kotak dialog untuk tempat pengisian kata kunci

Menentukan Sebelum Tengah Hari atau Sesudah Tengah Hari

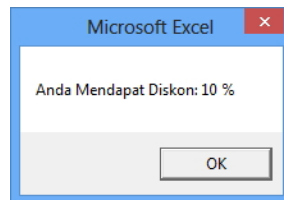
Dalam pengaturan waktu yang berlaku secara internasional, kita bisa mengamati bahwa pembagian waktu ditentukan berdasarkan "tengah hari" atau noon. Sebelum tengah hari disebut dengan istilah "pagi" sehingga wajar jika ada sapaan "Good Morning". Sedangkan setelah tengah hari disebut dengan istilah "afternoon" sehingga wajar jika ada sapaan "Good Afternoon". Nah di dalam sistem MS Excel, kita bisa melakukan pengecekkan waktu dengan menggunakan nilai 0.5. Jika waktu lebih kecil dari 0.5 maka dianggap masih pagi. Selanjutnya, jika waktu lebih besar dari 0.5 maka dianggap sudah sore.

Berikut penulisan script untuk menentukan apakah sekarang masih pagi atau sudah sore:

```
Sub CekWaktu()
 If Time < 0.5 Then MsgBox "Selamat Pagi"
 If Time >= 0.5 Then MsgBox "Selamat Sore"
End Sub
```

Tulisan yang akan muncul mengikuti informasi waktu yang dideteksi menggunakan script

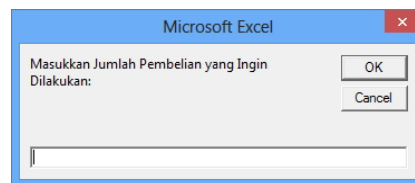
Sub Procedure untuk Menghitung Diskon

Mungkin sub procedure paling cocok untuk dunia bisnis, terutama retail, adalah sub procedure penghitung diskon. Ide dasarnya, si pengguna Macro ini memasukkan jumlah pembelian ke dalam kotak teks yang telah disediakan. Untuk jumlah-jumlah tertentu, kita bisa mengatur prosentase diskon yang akan diberikan kepada pembeli. Nah, besar kecilnya diskon itu tergantung banyak sedikitnya pembelian.

Berikut script yang digunakan untuk penghitungan diskon:

Tulisan yang menunjukkan besaran diskon akan sangat tergantung pada angka yang nanti Anda ketik ke dalam InputBox

```
Sub PotonganHarga()
 Dim Jumlah As Variant
 Dim Potongan As Double
 Jumlah = InputBox("Masukkan Jumlah Pembelian yang Ingin Dilakukan:
")
 Select Case Jumlah
  Case ""
   Exit Sub
  Case 0 To 25
   Potongan = 10
  Case 26 To 50
   Potongan = 20
  Case 51 To 100
   Potongan = 25
```

```
  End Select
  MsgBox "Anda Mendapat Diskon: " & Potongan & " %"
End Sub
```

Ketika sub procedure di atas dijalankan, maka akan muncul kotak dialog untuk menginput jumlah pembelian. Masukkanlah jumlah pembelian yang Anda inginkan ke dalam kotak teks yang telah disediakan itu.

Menentukan jumlah pembelian yang diharapkan

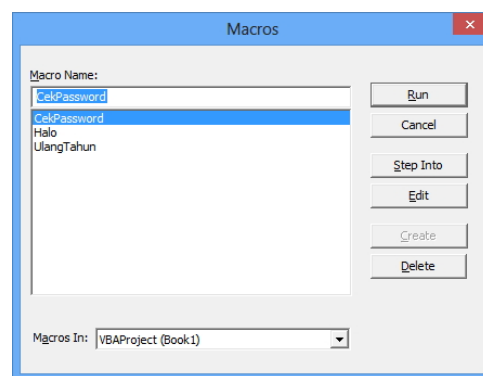
Cara Sederhana yang Mengupas Bagaimana Sub Procedure Dijalankan

Bisa jadi di dalam satu jendela Code terlihat beberapa sub procedure seperti gambar di bawah ini. Terlihat di dalam gambar, sub procedure-sub procedure itu dipisah oleh garis horizontal yang membedakan satu dengan lainnya.

```
(General)    CekPassword

Sub Halo()
    Dim Sapa _
    As String
    Sapa = _
    "Halo, Apa Kabar!"
    MsgBox Sapa
End Sub

Sub UlangTahun()
    If Date = #6/18/2013# Then
    MsgBox "Selamat Ulang Tahun Bro!"
    End If
End Sub

Sub CekPassword()
    UserName = InputBox("Tulis Password Anda: ")
    If UserName <> "Jubilee" Then GoTo Salah
    MsgBox "Login Berhasil. Selamat Datang Jubilee"
    Exit Sub
Salah:
    MsgBox "Maaf. Password yang Anda masukkan salah"
End Sub
```

Satu jendela Code yang memuat beberapa sub procedure berbeda-beda

Untuk menjalankan satu sub procedure, maka lakukan langkah-langkah sederhana berikut ini:

1. Klik di dalam sub procedure yang ingin dijalankan. Dengan demikian, posisi kursor berada di dalam sub procedure itu.
2. Tekanlah tombol **F5**. Seharusnya, sub procedure itu akan langsung dijalankan. Namun jika muncul jendela **Macros** seperti gambar di bawah ini, maka pilihlah nama sub procedure yang ingin dijalankan.

Jendela Macros yang bisa dipakai untuk memilih sub procedure yang ingin dijalankan

3. Tekan tombol **Run**. Selanjutnya, Anda akan melihat sub procedure yang telah dijalankan itu.

Membuat Function Sederhana

Berbeda dengan sub procedure, Function berfungsi untuk menghasilkan data atau nilai tertentu. Oleh karena itulah, di dalam pembuatan function paling tidak ada keterlibatan argument (data-data yang akan diolah) dan rumus-rumus yang berfungsi untuk mengolah argument itu.

Selanjutnya, function tersebut akan menghasilkan sebuah nilai atau data olahan yang bisa kita masukkan ke dalam worksheet melalui Formula Bar.

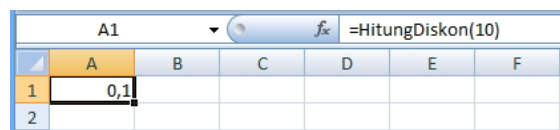
Berikut contoh-contoh pembuatan function sederhana untuk mengatasi masalah-masalah bisnis sehari-hari:

Menghitung Diskon Menggunakan Function

Pada contoh sebelumnya, proses penghitungan diskon dilakukan dengan memanfaatkan Sub Procedure. Kali ini, kita akan melakukan hitung-hitungan yang sama namun dengan mengandalkan function. Tulislah script di bawah ini di dalam jendela Code:

```
Function HitungDiskon(jumlahbarang)
 Dim jumlah As Integer
 Dim Potongan As Double
 Select Case jumlahbarang
  Case 0 To 25
   Potongan = 10 / 100
  Case 26 To 50
   Potongan = 20 / 100
  Case 51 To 100
   Potongan = 25 / 100
 End Select
 HitungDiskon = Potongan
End Function
```

Berikut cara memanfaatkan function di atas dalam sebuah worksheet:

1. Masuklah ke dalam worksheet MS Excel terlebih dulu.
2. Di dalam formula bar, tulislah function sebagai berikut: **=HitungDiskon(10)**.

Menulis function di dalam Formula Bar untuk mendapatkan hasil di dalam salah satu sel yang sedang aktif

3. Tekanlah tombol **Enter** dan di dalam sel yang saat itu aktif akan tercatat nilai **0,1**.

Setelah menghasilkan nilai di dalam sebuah sel, maka hitung-hitungan tersebut bisa diolah menggunakan fungsi atau formula lain yang mungkin Anda tulis di dalam sel-sel yang berlainan.

Menggabungkan Function dan Sub Procedure

Kita sudah membuat dua variasi penghitung diskon. Yang pertama adalah dengan menggunakan sub procedure saja, dan yang lainnya berbentuk function saja. Sebenarnya, untuk menghemat script, kita bisa menyatukan function ke dalam sub procedure sehingga nanti hasil yang didapat bisa dialihkan ke dalam MsgBox. Caranya pertama-tama, tulis function di atas terlebih dulu. Lantas, lanjutkan dengan menulis sub procedure seperti gambar di bawah ini:

```
Sub MencariDiskon()
 Dim jumlah As Integer
 Dim PersenDiskon As Double
 jumlah = InputBox("Masukkan jumlah pembelian: ")
 PersenDiskon = HitungDiskon(jumlah)
 MsgBox "Anda Mendapat Diskon: " & PersenDiskon
End Sub
```

Karena berbentuk sub procedure, maka tekanlah tombol **F5** untuk menjalankan script di atas. Nanti, Anda akan melihat hasil seperti ketika menulis sub procedure pertama. Yang membedakan adalah “jeroan” script di mana pada script di atas, rumus penghitung diskon kita letakkan di dalam function yang terpisah.

Hanya saja, script di atas tidak menghasilkan data atau nilai yang ditulis di dalam sebuah sel. Untuk membuat ada penulisan data atau nilai di dalam sebuah sel, maka tambahkanlah satu baris script di bawah ini sebelum End Sub:

```
    Range("A1").Value = PersenDiskon
End Sub
```

Pada potongan script di atas, hasil penghitungan function akan diletakkan ke dalam sel A1. Jika Anda ingin agar hasil tersebut diletakkan di sel mana pun yang saat itu aktif, maka ubahlah satu baris di atas menjadi:

```
    ActiveCell.Value = PersenDiskon
End Sub
```

Jadi, sel yang akan diberi hasil penghitungan tidak terikat pada sel A1, tapi pada sel yang saat itu sedang aktif (terseleksi).

MEMBUAT FUNCTION PRIBADI UNTUK MS EXCEL

MS Excel memiliki ratusan function built-in yang siap-pakai. Beberapa di antaranya pasti sering Anda gunakan seperti SUM, SUMIF, IF, COUNT, dan lain sebagainya. Jika menggunakan MS Excel 2013, maka Anda bisa memakai hampir dengan 450 function.

Nah kalau tidak ada function yang benar-benar praktis atau bisa mengatasi kebutuhan Anda sendiri, maka lewat contoh di atas, Anda bisa membuat function sendiri. Anda bisa membuat function untuk menghitung pajak, bunga, tingkat produktivitas pegawai, dan lain sebagainya.

Apa saja kelebihan dan kekurangan function buatan sendiri? Berikut kelebihannya:

- ***Bisa menyederhanakan formula**. Pada dasarnya, fungsi-fungsi yang telah disediakan oleh MS Excel sudah cukup mampu mengatasi masalah kita dalam bisnis sehari-hari. Namun semakin kompleks masalah yang kita hadapi, maka fungsi-fungsi itu harus digabung dengan pernyataan lain sehingga menghasilkan formula yang panjang. Kehadiran fungsi buatan sendiri yang disusun lewat VBA bisa mengatasi masalah kompleksitas formula yang harus ditulis di dalam Formula Bar.*
- ***Membuat worksheet menjadi "unik":** Dengan banyaknya fungsi-fungsi asing yang akan ditemui oleh para pegawai, menyebabkan worksheet dan workbook secara umum menjadi tampak 'unik' dan bersifat praktis. Dengan demikian fungsi-fungsi ciptaan sendiri itu akan lebih mudah digunakan oleh para pegawai yang tidak begitu melek MS Excel.*

- ***Fungsi-Fungsi itu dapat dijual:*** *Kalau Anda bisa menciptakan fungsi yang benar-benar bagus dan mampu menyelesaikan masalah kompleks, maka ada peluang bagi Anda untuk menjual fungsi itu kepada pihak lain.*

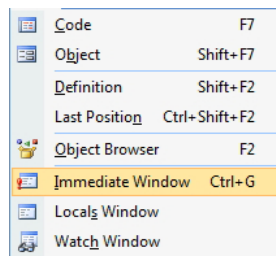
Sedangkan kekurangan fungsi buatan sendiri adalah:

- ***Secara umum harus tahu VBA:*** *Karena fungsi dibuat menggunakan VBA, maka Anda harus menguasai VBA.*
- ***Lebih lambat diproses oleh MS Excel:*** *Dibandingkan fungsi built-in, fungsi-fungsi buatan sendiri yang disusun menggunakan VBA lebih lambat diproses oleh MS Excel.*

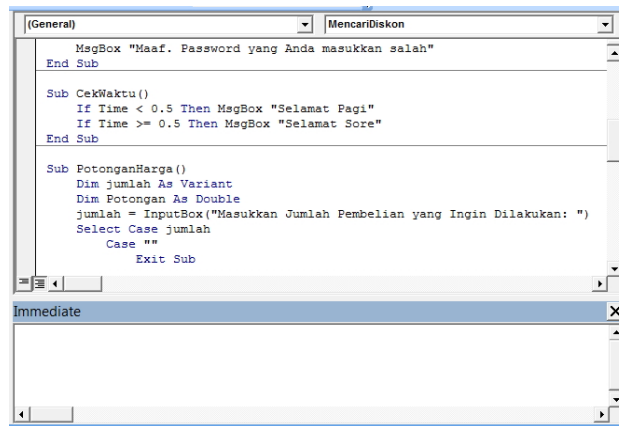
Menggunakan Immediate Window

Immediate Window merupakan fitur yang telah disediakan di dalam jendela Visual Basic Editor untuk membantu Anda menguji bagaimana sub procedure itu melakukan perintah tertentu sebelum kita benar-benar menggunakannya di dalam workbook.

Immediate Window dapat dimunculkan dengan salah satu dari dua alternatif berikut:

- Pilih menu **View > Immediate Window**.

Memilih Immediate Window untuk menampilkan jendela tersebut

- Tekan tombol shortcut **Ctrl+G.**

Dari dua metode di atas, akan terlihat tampilan Immediate Window yang secara default diletakkan di bawah panel Code:

Panel Immediate di bawah panel Code

Contoh Penggunaan Immediate Window

Panel Immediate Window akan membantu kita memvisualisasikan hasil yang akan terlihat ketika sebuah sub procedure dijalankan. Berikut contoh penerapannya:

1. Pertama-tama, buatlah sub procedure dengan mengetikkan script seperti berikut ini:

```
Sub UbahKapital()
 Dim Teks As String
  Teks = "Namaku si Boy"
  Teks = UCase(Teks)
 Debug.Print Teks
End Sub
```

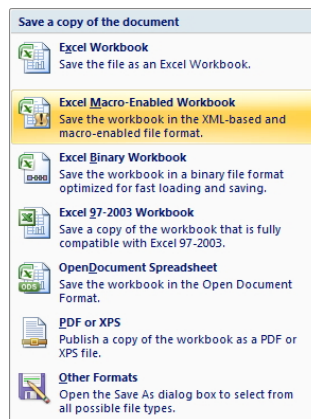
2. Lantas, pindahlah ke dalam Immediate Window dengan mengklik di dalam panel tersebut.
3. Ketik nama sub procedure di atas, dalam hal ini: **UbahKapital**.
4. Tekan tombol **Enter** dan nanti akan terlihat hasil dari pengolahan sub procedure di atas.

Panel Immediate untuk menguji sub procedure

Menyimpan dan Membuka Workbook yang Mengandung VBA

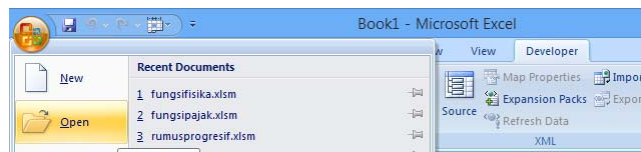
Ketika Anda bekerja dengan menggunakan workbook yang mengandung VBA di dalamnya, maka proses penyimpanan workbook itu menjadi sebuah file tidaklah sama jika dibandingkan proses penyimpanan workbook tanpa VBA.

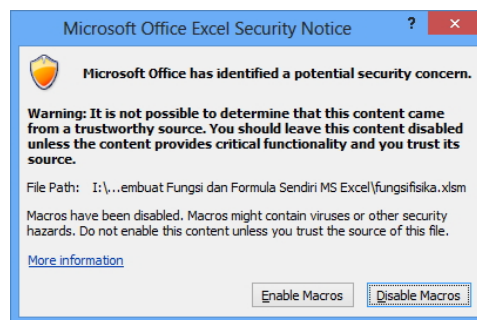
Untuk menyimpan workbook yang mengandung VBA, lakukan langkah-langkah sebagai berikut:

1. Tekan tombol **Office** dan sorot pada pilihan **Save As**.
2. Klik pada pilihan **Excel Macro-Enabled Workbook**.

Pilih opsi Excel Macro-Enabled Workbook

3. Tulislah nama file yang diinginkan. Coba cek, ketika Anda memilih opsi di atas, maka ekstensi file workbook itu adalah **.xlsm**.
4. Tekanlah tombol **Save**.

Sedangkan proses membuka file workbook yang mengandung VBA (Macro) pun cukup istimewa dibandingkan dengan file workbook biasa. Berikut prosesnya:

1. Tekan tombol **Office** dan klik pada pilihan **Open**.

Memilih opsi Open untuk membuka file workbook ber-Macro

2. Pilih file workbook tersebut seperti biasa dan tekan tombol **Open**.
3. Nah, kalau file itu mengandung VBA maka umumnya akan terlihat kotak dialog seperti gambar di bawah ini. Untuk membuka file workbook itu, tekan tombol **Enable Macros**.

Tombol Enable Macros untuk mengaktifkan fitur VBA

4. File workbook ber-VBA itu pun akan terbuka di hadapan Anda.

Menyelesaikan Berbagai Bisnis Menggunakan VBA

Ada banyak masalah bisnis yang bisa diselesaikan menggunakan VBA, entah lewat sebuah function, sub procedure sederhana, hingga aplikasi yang melibatkan UserForm. Untuk membantu Anda memahami secara lebih jelas bagaimana caranya menyelesaikan berbagai masalah bisnis

memakai VBA, maka di dalam bab ini akan dijelaskan berbagai contoh kasus yang menarik untuk dicoba. Anda bisa memodifikasinya sendiri sesuai kebutuhan bilamana contoh kasus-contoh kasus yang dibahas di dalam bab ini menyerupai permasalahan yang saat ini Anda hadapi.

Mewarnai Sel untuk Melihat Data-Data Angka dengan Kriteria Tertentu

Sel-sel yang ada di worksheet bisa diwarnai untuk menandai data-data yang ada di dalamnya berdasarkan kriteria-kriteria tertentu. Sebagai contoh, Anda bisa membuat agar sel itu berwarna merah jika mengandung angka melebihi/lebih sedikit dari jumlah tertentu. Dengan demikian, Anda bisa mengecek mana penjualan yang kurang dari target atau mana pengeluaran yang melebihi budget yang telah ditetapkan.

Caranya adalah sebagai berikut:

1. Masuklah ke dalam Visual Basic Editor dan buatlah **Module** baru.
2. Ketik script seperti di bawah ini:

```
Sub WarnaNegatif()
 Dim Sel As Range
 If TypeName(Selection) <> "Range" Then Exit Sub
 Application.ScreenUpdating = False
 For Each Sel In Selection
  If Sel.Value > 999999 Then
   Sel.Interior.Color = RGB(255, 0, 0)
  Else
   Sel.Interior.Color = -4142
  End If
 Next Sel
End Sub
```

3. Sekarang, kembalilah ke dalam worksheet.

4. Buatlah tabel seperti contoh di bawah ini (kasus pembayaran listrik bulanan).

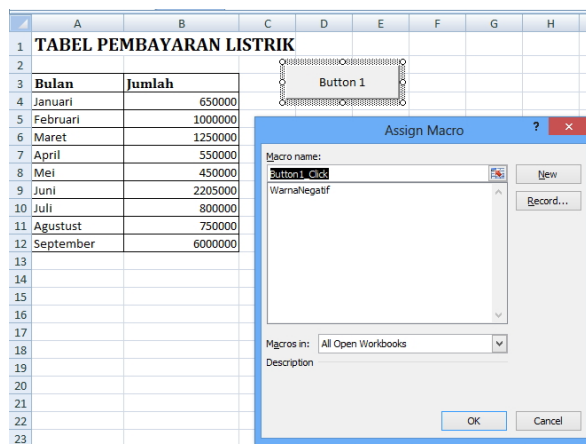
TABEL PEMBAYARAN LISTRIK

Bulan	Jumlah
Januari	650000
Februari	1000000
Maret	1250000
April	550000
Mei	450000
Juni	2205000
Juli	800000
Agustust	750000
September	6000000

Membuat tabel yang mendata besaran tarif bulanan untuk listrik

5. Klik tab **Developer** dan buatlah tombol dengan mengklik ikon **Insert > Button**. Nanti jika muncul jendela **Assign Macro**, klik **WarnaNegatif**.

TABEL PEMBAYARAN LISTRIK

Bulan	Jumlah
Januari	650000
Februari	1000000
Maret	1250000
April	550000
Mei	450000
Juni	2205000
Juli	800000
Agustust	750000
September	6000000

Membuat tombol untuk memicu sub procedure WarnaNegatif

6. Seleksi sel B4 sampai B12. Kalau sudah, tekan tombol itu dan dalam sekejap, sel-sel yang terseleksi itu dan yang memiliki data angka lebih dari 999999 akan ditandai dengan warna merah pada seluruh interior sel.

TABEL PEMBAYARAN LISTRIK

Bulan	Jumlah
Januari	650000
Februari	1000000
Maret	1250000
April	550000
Mei	450000
Juni	2205000
Juli	800000
Agustust	750000
September	6000000

Sel-sel akan ditandai dengan warna merah ketika memiliki angka di atas 999999

Agar sel-sel berisi angka-angka yang tidak masuk kriteria tetap seperti kondisi semula, maka gunakan konstanta -4142 untuk membuatnya dalam keadaan No Fill.